

Eina Web per a la Consulta i la Visualització d'Imatges

Sergio Moral Gutiérrez

Resum—Aquest projecte servirà per a visualitzar les imatges d'una base de dades d'imatges d'escenes sintètiques que l'equip de color del Centre de Visió per Computador (CVC, Universitat Autònoma de Barcelona) ha desenvolupat com a part de la seva recerca. Al ser una aplicació web, l'usuari no necessita un software concret i qualsevol plataforma li permetrà consultar la informació. Per a realitzar aquest projecte s'utilitzen les tecnologies, tècniques i eines més extenses i populars en el desenvolupament web, tals com PHP, AJAX, jQuery, MySQL, Javascript, HTML i CSS. Aquesta eina està dirigida a un usuari concret que estigui interessat en la producció d'aquestes imatges sintètiques d'escenes.

Paraules clau— PHP, AJAX, jQuery, MySQL, Javascript, HTML, CSS, aplicació web, eina web, desenvolupament, imatges, sintètiques, escenes, base de dades, DOM, XML.

Abstract—This project will serve to view images from a database of images of synthetic scenes that the color team of the Computer Vision Centre (CVC, Universitat Autònoma de Barcelona) has developed as part of their research. Being a web application, user does not need any specific software and any platform will enable him/her to consult the information. For this project the most widespread and popular technologies, techniques and tools are used in web development such as PHP, AJAX, jQuery, MySQL, Javascript, HTML and CSS. This tool is aimed at a specific user who is interested in the production of these synthetic images of scenes.

Index Terms— PHP, AJAX, jQuery, MySQL, Javascript, HTML, CSS, web application, web tool, development, images, synthetic, scenes, database, DOM, XML.

1 INTRODUCCIÓ

A vui en dia Internet permet posar continguts a l'abast de molta gent, divulgar informació i coneixements, donar accés a dades, etc. És un mitjà molt utilitzat per transmetre dades científiques i recursos per recerca, tant investigadors com estudiants accedeixen a Internet per trobar allò que necessiten de manera ràpida i senzilla.

En visió per computador, es treballa amb imatges i per tant, el que es comparteix moltes vegades són conjunts d'imatges per testear algorismes. En els darrers anys, aquests conjunts de dades són molt grans i inclouen moltes imatges (milers). Per això es necessiten eines que permetin visualitzar i seleccionar aquestes imatges.

Un recurs molt habitual són imatges que de forma individual consumeixen un ample de banda i un cost computacional que en la gran majoria de casos no suposen un problema.

Ara bé, quan en una consulta s'han de mostrar una gran quantitat d'aquestes imatges, el consum del ample de banda i la necessitat de recursos creixen exponencialment. Es pot consultar un exemple en la web d'una base de dades de textures de la Universitat de Oxford [1] on les imatges triguen en carregar encara que el seu tamany ha sigut reduït per estalviar en el cost que comporta.

El grup de color del CVC ha creat un d'aquests conjunts d'imatges amb 68.220 imatges d'escenes que serveixen per avaluar algorismes de diferents camps de la visió per computador. A partir de 10 escenes base s'han generat imatges variant una sèrie d'atributs:

- Punts de vista: Cada escena té tres punts de vista diferents, amb tres nivells de zoom cadascun, obtenint així nou punts de vista diferents.
- Materials: Per a cada escena hi ha tres permutacions diferents de reflectància pels materials representats (figura 1).
- Il·luminant: Hi ha sis il·luminants diferents entre ells (un il·luminant blanc, tres il·luminants plankians i dos il·luminants no plankians). Per cada escena es generen imatges amb un il·luminant ("single il·luminant") i amb dos il·luminants ("multi il·luminant").

• E-mail de contacte: smoral87@gmail.com
• Menció realitzada: Enginyeria del Software
• Treball tutoritzat per: Robert Benavente Vidal
• Curs 2014/15



Figura 1. Escena amb diferents materials. [2]

Per comprendre millor aquesta estructura, en la figura 2 es pot veure la representació de com es troba emmagatzemat al disc dur. Totes les imatges es troben ubicades en carpetes que representen els atributs anteriorment esmentats.

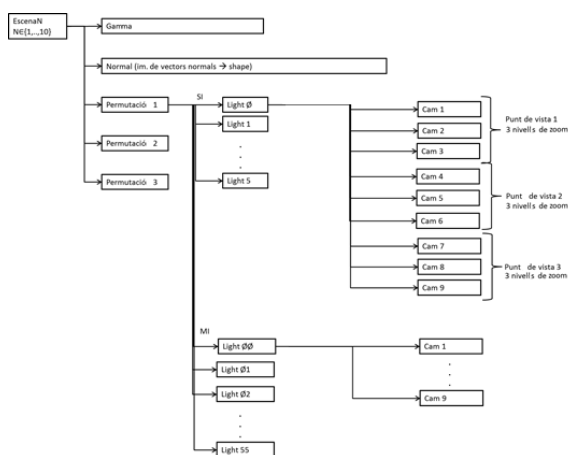


Figura 2. Estructura de directoris on s'emmagatzemaven originalment les dades al disc. El diagrama només representa els directoris de permutació 1, però es repeteix la mateixa organització per les permutacions 2 i 3.

Aquesta base de dades es part de la investigació del grup de color en el Centre de Visió per Computador (CVC) i es vol permetre l'accés a les dades des de la seva web <http://www.cic.uab.cat/> mitjançant una aplicació que permeti a qualsevol investigador, d'aquest grup de recerca, consultar i visualitzar les imatges de les escenes.

El principal problema que s'identifica és que la base de dades consta de un gran volum de imatges, i això és una càrrega molt alta a l'hora de visualitzar-les en la web. Donat aquest problema, existeix la necessitat de construir una aplicació web que sigui ràpida, dintre dels límits del ample de banda, visualitzant les imatges i executant les consultes de manera òptima.

La resta del document s'organitza de la següent forma. A la secció 2 es revisa l'estat actual de les tecnologies de desenvolupament web. A continuació, a la secció 3 es tracten els objectius i requisits, separant els funcionals dels no funcionals. A la secció 3 es detalla la metodologia, seguida en el desenvolupament de la base de dades,

l'eina web i la seva integració i instal·lació final. A la secció 4 s'exposen els resultats del projecte així com les característiques de l'eina. Als resultats li segueixen les conclusions a la secció.

2 ESTAT DE L'ART

Internet ens posa a l'abast una quantitat enorme d'informació que es veu representada en les pàgines web. Quan es navega a Internet, com a usuari, es pot trobar des de pàgines web senzilles que només mostren informació, fins a pàgines web que interactuen rebent i enviant dades.

Les aplicacions web, o eines web, es diferencien de les pàgines web pel fet d'executar codi, interpretar-lo i poder treballar amb dades des de un servidor extern. Amb aquesta capacitat, una aplicació web té l'avantatge de poder realitzar tasques més concretes sense la necessitat d'adquirir un software adicional. També qualsevol persona pot accedir des del seu ordinador, mòbil o tablet sense requisits de hardware.

Des de fa molt de temps existeixen eines web que funcionen sobre una base de dades a la que fan consultes per rebre o enviar dades. Concretament podem trobar pàgines web amb informació de qualsevol tipus i imatges associades a aquesta informació. Aquesta informació pot ser variable, donats uns filtres de selecció l'usuari pot decidir què vol veure o com ho vol veure.

Actualment existeixen multitud de tecnologies web. Com a tecnologia web entenem tot el conjunt d'eines que faciliten trobar millors resultats a l'hora de desenvolupar una web. Trobem navegadors potents i extensos a diferents plataformes (Chrome, Firefox, Opera, etc.) que ens permeten guardar una única configuració a un servidor extern (cloud) i navegar des d'un tablet fins a un ordinador. Aquests navegadors es comuniquen amb un servidor HTTP on li envien pàgines web. Aquestes pàgines web es poden programar amb diferents llenguatges de programació web i s'executen al servidor o al navegador, depenent del llenguatge. Tot aquest ecosistema és l'estat actual de les tecnologies web i la avantatge que tenen pels desenvolupadors és que es pot implementar en el llenguatge que es desitgi ja que directe o indirectament existeixen connectors que permeten al servidor HTTP interpretar el codi.

Eines com HTML5 estan a l'ordre del dia, i organitzacions grans l'estan incorporant en les seves aplicacions web per substituir altres eines obsoletes. Encara que no hi ha un grup d'eines o tecnologies predominants a l'hora de desenvolupar aplicacions web, algunes com Active Server Page (ASP) de Microsoft, Java (JavaServer Pages, Applets) d'Oracle o PHP de PHP Group són de les favorites, pels desenvolupadors, a l'hora d'escollir un llenguatge de programació a la banda del servidor.

3 OBJECTIUS I REQUISITS

3.1 Objectius

L'objectiu principal d'aquest treball és desenvolupar una eina web per a visualitzar imatges d'un conjunt d'imatges d'escenes sintètiques. Per a la visualització hi ha d'haver diferents opcions que permetin a qualsevol usuari interessat poder escollir quines imatges vol consultar. Aquesta eina ha de ser ràpida i eficient a l'hora de fer les seves funcions i s'ha de instal·lar a la web del equip de color del CVC.

L'objectiu secundari és estendre l'eina web per permetre ampliar la base de dades. D'aquesta manera l'eina ha d'actualitzar el contingut dels atributs en cas de afegir-hi nous.

3.2 Requisits

Al principi del projecte es van concertar reunions amb el tutor (client) per perfilar millor els requisits relacionats amb el contingut, estructura i usabilitat de l'aplicació web. A partir d'aquestes reunions, es va fer l'anàlisi de requisits del projecte. La llista de requisits funcionals i no funcionals que es van determinar es llista a continuació:

Requisits funcionals:

- RF1.** L'eina web ha de ser capaç de mostrar les imatges de la base de dades.
- RF2.** L'eina web ha de tenir tipus de filtres de selecció per visualitzar les imatges.
- RF3.** L'usuari ha de poder seleccionar més d'una escena per visualitzar imatges.
- RF4.** L'usuari ha de poder seleccionar la quantitat d'imatges desitjades per pàgina.
- RF5.** L'usuari ha de poder trobar imatges amb pocs clics.

Requisits no funcionals

- RNF1.** L'eina web ha de ser implementada en la web de l'equip de color del Centre de Visió per Computador (CVC).
- RNF2.** El disseny de la interfície ha de ser el mateix que el de la web del equip de color del CVC.
- RNF3.** Els filtres de selecció s'han d'agrupar per tipus en la base de dades.
- RNF4.** L'eina web ha de tenir un tamany adequat per resolucions de pantalla grans.
- RNF5.** L'estructura del contingut visualitzat ha de ser agradable visualment.
- RNF6.** L'eina web ha de ser ràpida carregant les imatges.
- RNF7.** L'eina web ha de ser compatible amb navegadors populars.
- RNF8.** La base de dades ha de ser escalable per a futures ampliacions.
- RNF9.** L'eina web ha de ser fàcil d'utilitzar.

4 METODOLOGIA

El desenvolupament del projecte s'ha dut a terme seguint una planificació que es pot trobar a l'annex.

4.1 Disseny general

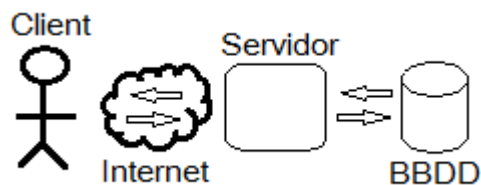


Figura 3. Representació de l'eina web. L'eina es troba al servidor i quan un usuari fa una petició de consulta a través de Internet, el servidor fa una consulta amb les dades de la petició del client a la base de dades. La base de dades retorna els resultats d'aquesta consulta al servidor i aquest mostra al client els resultats.

El projecte inclou dues tasques principals i s'utilitza una metodologia en cascada amb algunes variacions. Es proposen fites o tasques que s'han de complir realitzant les tasques necessàries per aconseguir-lo.

Les dues tasques principals són:

- Creació de la base de dades.
- Desenvolupament de l'eina web.

4.2 La base de dades

Inicialment, les imatges de cada escena es trobaven ordenades mitjançant una estructura de directoris (veure figura 2). Per optimitzar el rendiment de la base de dades, i disposar de totes les imatges en un únic directori centralitzat, es va generar un script que copia a una nova ubicació les imatges i els canvia el nom per un de nou en el qual es pot identificar els atributs de cada imatge, amb el propòsit de reubicar les dades en una carpeta i utilitzar una nomenclatura concreta.

4.2.1 Disseny de la BBDD

Per a dissenyar la base de dades es varen plantejar diversos diagrames d'entitat-relació. Es va optar per utilitzar una única entitat per a evitar l'ús de joins, els quals tenen un cost computacional elevat [3], i crear un índex per escena que es el major filtre aplicable. Amb aquest criteri, el disseny de la base de dades es el següent:

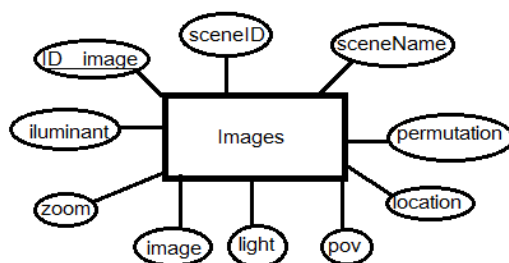


Figura 4. Diagrama E-R de la base de dades.

Per tant, la base de dades queda dissenyada de la següent forma:

Taula 1. Atributs de la entitat Images.

Images	
ID_Image	INT (UNSIGNED; AUTO_INCREMENT; PRIMARY KEY)
iluminant	VARCHAR(2) (NOT NULL)
sceneID	SMALLINT(3) (UNSIGNED; NOT NULL)
sceneName	VARCHAR(30) (NOT NULL)
permutation	SMALLINT(3) (UNSIGNED)
light	SMALLINT(3) (UNSIGNED)
pov	SMALLINT(3) (UNSIGNED; NOT NULL)
zoom	SMALLINT(3) (UNSIGNED; NOT NULL)
location	VARCHAR(100) (NOT NULL)
Image	VARCHAR(15) (NOT NULL)
Indexs	
sceneIDX	(sceneID)

Per accedir a la base de dades es crea un usuari que es el que usa l'aplicació web per fer les consultes a la base de dades. A aquest usuari només se li atorguen permisos de selecció ("select") que és el permís necessari per fer les consultes.

D'aquesta manera assegurem que si hi ha algun usuari malintencionat que vulgui aprofitar vulnerabilitats a la base de dades no podrà fer modificacions.

4.2.2 Eines utilitzades

Per generar l'script mencionat a l'apartat 4.2 s'ha utilitzat Matlab [4]. És un entorn de programació integrat molt útil, entre altres coses, per fer aquests tipus de scripts de manera fàcil i ràpida.

A l'hora de crear la base de dades s'ha fet en MySQL [5], una eina molt popular i estesa, sense cost, que té un gran suport per part de la comunitat a Internet. Tot això fa que hi hagi recursos més que suficients per poder solucionar possibles problemes durant el desenvolupament o la integració.

4.2.3 Implementació

Les imatges es troben a un directori amb una nomenclatura específica a la que s'especifiquen els atributs. Per exemple: M01_P0_L0_POV0_Z0_gamma.png on

- M correspon al il·luminant múltiple.
- 01 al id de l'escena.
- P0 a la combinació 0.
- L0 a la llum 0.
- POV0 i Z0 corresponen al punt de vista 0 amb un nivell de zoom 0.
- Gamma correspon a la imatge.

Mitjançant scripts es generen els fitxers SQL necessaris per introduir tots els atributs de cada imatge a la base de dades.

Per últim, per línia de comandes de MySQL s'insereixen les dades de cada imatge que es troben als fitxers SQL generats anteriorment, omplint la base de dades amb tota la informació de les imatges.

4.2.4 Proves

Per provar que el contingut de la base de dades estigués bé s'han fet proves de caixa negra on s'especificava una entrada de dades (en aquest cas unes consultes en llenguatge SQL) i s'esperava que el resultat fos un grup de valors de sortida (registres de la base de dades). Aquest grup de valors de sortida es correspon amb l'estructura inicial de les imatges (veure figura 2). Donat aquest tipus de proves, s'ha provat que per a les consultes per a cada tipus d'atribut, el resultat fos l'esperat.

També s'ha consultat que el nombre total de registres coincidís amb el total d'imatges que hi ha a la ubicació on es troben totes les imatges, provant així que no falta cap registre a la base de dades.

Per a cada atribut s'ha comprovat que a la base de dades hi hagi el mateix nombre de registres assegurant la integritat i la fiabilitat de les dades.

4.3 L'eina web

L'eina es comunica amb la base de dades MySQL per visualitzar les imatges que el client vol consultar.

4.3.1 Disseny de la web

Per dissenyar l'eina es van plantejar diferents prototips dels que el client va fer un procés d'acceptació. El disseny acceptat inclou un menú lateral a l'esquerra que conté els filtres de selecció, un contenidor a la dreta del menú on es carrega la informació que el client vol consultar, un peu de pàgina amb informació rellevant del projecte i una capçalera amb diversos enllaços d'interès.

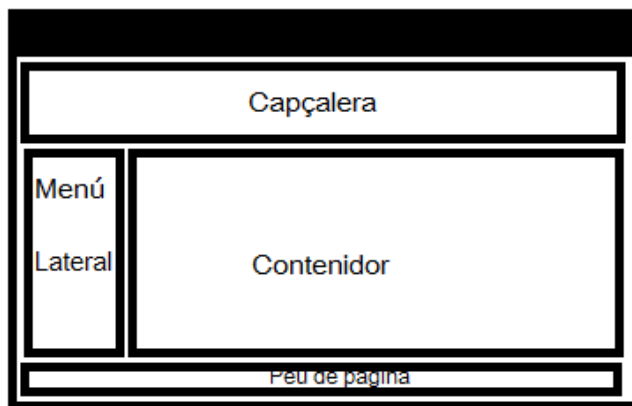


Figura 5. Esquema de l'eina web.

4.3.2 Eines utilitzades

El desenvolupament de l'eina es basa en HTML i PHP [6]. No obstant, per agilitzar la visualització de les imatges i reduir la càrrega en el servidor, algunes de les funcions s'implementen en AJAX [7] utilitzant jQuery [8].

AJAX és una tècnica de Javascript que permet actualitzar el contingut d'una pàgina web (tot o una part) amb informació proporcionada per un servidor o un altre document HTML. L'intercanvi d'informació amb el servidor es realitza de forma asíncrona per la qual cosa es pot actualitzar només una part del contingut (ja sigui en un element concret del codi HTML o PHP, o una part major de la pàgina com pot ser tot el contingut de un contenidor) sense carregar la resta del document. Per tant, amb AJAX es pot optimitzar la càrrega de la pàgina en l'eina web. A la figura 6 es pot veure un esquema del funcionament de AJAX.

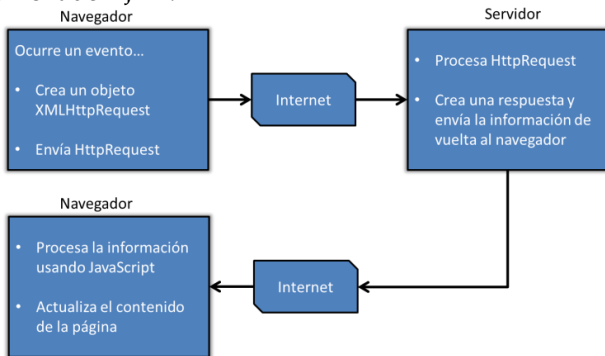


Figura 6. Esquema del funcionament d'AJAX [7] des de que es produeix un event fins que s'obté la resposta del servidor amb el contingut nou per actualitzar la pàgina.

El motiu d'utilitzar aquestes tecnologies i tècniques és el mateix que amb la base de dades en MySQL, es deu a que son lliures i tenen un gran suport en diferents mitjans que ajuden a solucionar els problemes durant el desenvolupament. Són eines molt esteses i populars, no tenen cost econòmic i garanteixen superar situacions crítiques.

Per simplificar el procés de desenvolupament en Javascript, existeix la llibreria jQuery que té funcions d'alt nivell per interactuar amb el Document Object Model (DOM) [10], facilitant la programació i donant suport a diferents navegadors.

DOM és una interfície de programació d'aplicacions (API) que proporciona estàndards d'objectes per a poder ser representats a HTML i XML. A través de DOM, les aplicacions poden accedir i modificar el contingut, estructura i estil dels documents HTML i XML. En aquest projecte, DOM es la interfície que utilitza jQuery per visualitzar el contingut que retorna el servidor en la pàgina PHP de l'eina implementada.

Per garantir els requisits s'utilitza un plugin de Javascript anomenat "Lazy Load Plugin for jQuery" [9] que redueix el cost computacional que comporta carregar una gran quantitat d'imatges en una pàgina web. Lazy Load endarrereix la càrrega d'imatges fent que aquestes que són fora del viewport no es carreguin fins que l'usuari es situa a elles. Es el punt oposat a la precàrrega d'imatges que es fa de manera genèrica.

4.3.3 Implementació

Com es pot veure a la planificació (veure annex), es va construir un esquelet en HTML al qual es defineixen les parts de l'aplicació web corresponents al menú que conté els filtres de selecció, la capçalera, el contenidor de la informació i el peu de pàgina, tal com s'ha descrit al punt 4.3.1.

Al menú lateral que correspon als filtres de selecció es programa mitjançant PHP per que consulti a la base de dades tots els atributs disponibles i així es genera amb HTML un formulari de checkboxs per a que l'usuari filtri la informació que vol veure.

Aquest formulari fa ús de funcions AJAX per convertir en llistes les opcions marcades per a cada atribut i retornar-les a un nou fitxer "contenidor" on es tracten les dades.

Aquest contenidor substitueix el que havia originalment a la dreta del menú lateral i és on s'extreuen les dades que envia el formulari per fer les consultes a la base de dades. Aquestes consultes retornen les imatges que corresponen a la selecció que ha fet l'usuari i se'ls aplica el Lazy Load, cridant a les funcions del plugin.

Per a les imatges que es mostren en aquest contenidor es crea una taula per controlar la quantitat que hi surten per files per assegurar la integritat del disseny. Sempre tenen un tamany fix, encara que es crea un enllaç a una finestra nova on visualitzar la imatge a tamany complet.

Totes les consultes que l'usuari fa es poden comprimir i descarregar.

4.3.4 Proves

Es van fer proves d'unitat per als mòduls del menú lateral i contingut (o contenidor) de l'eina web. En el cas del menú lateral, donat que envia informació al mòdul del contingut, es va provar que els valors d'entrada (checkboxs seleccionats) corresponguessin amb els valors de sortida (informació que s'envia en forma de llista).

Al mòdul del contingut es va provar que els valors d'entrada (llista amb diferents valors) i els valors de sortida (informació en forma de text) fossin els esperats.

Un cop es varen fer les proves d'unitat es va procedir a fer les proves d'integració entre els dos mòduls. Es va provar que les dades d'entrada al mòdul del menú lateral fossin les dades de sortida al mòdul del contingut. Les següents figures mostren les proves d'integració entre els dos mòduls:

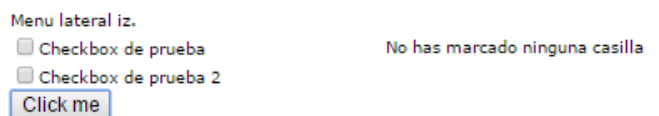


Figura 7. Eina web. No mostra cap resultat (dades de sortida) ja que no hi ha dades d'entrada (checkboxs seleccionats).

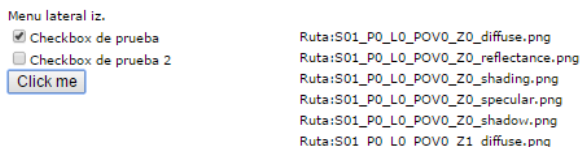


Figura 8. Eina web. Es mostren els resultats (dades de sortida) que corresponen al checkbox 1 (dades d'entrada) seleccionats.



Figura 9. Eina web. Es mostren els resultats (dades de sortida) que corresponen al checkbox 2 (dades d'entrada) seleccionats. Es pot veure com les dades del checkbox 1 no apareixen.

En el cas del mòdul del contingut, es va provar que l'espai per les imatges fos l'adequat. També es va provar que al sortir molt contingut la pàgina s'expandís per auto adaptar-se.



Figura 10. Eina web. Ajust automàtic del contingut.



Figura 11. Eina web. Ajust automàtic del contingut 2.

4.4 Integració

Una vegada implementades les dues parts principals del projecte, es va integrar la base de dades sota l'aplicació web. Per fer-ho es van comprovar els codis de programació en PHP que connecten amb la base de dades i el retorn de les consultes.

A més, mitjançant codi PHP, es programa un sistema de paginació del contingut. Aquest sistema consisteix en fer la mateixa consulta dos cops. A simple vista pot semblar que és malbaratar recursos, però és necessari per tal d'estalviar recursos, ja que la primera consulta es fa per

determinar la quantitat total de registres que es mostraran.

Aquests nombre de registres, juntament amb el tamany de pàgina establert s'utilitza per calcular el nombre màxim de pàgines i així, a la segona consulta, establir l'inici del registre on mostrar continguts.

Per exemple, si mostrem 50 elements per pàgina (tamany de pàgina) i l'inici és la pàgina 1, amb el paràmetre límit $[(1-1),50]$ al final de la segona consulta fa que només retorni 50 elements començant per la posició 0. Quan es canviï de pàgina, tornarà a calcular el tamany i haurà de ser el mateix per mantenir la integritat.

Així, es pot fer un menú de paginació al final del contingut per a que l'usuari pugui navegar per les diferents pàgines.

Per últim, es va afegir un menú desplegable que permetés a l'usuari escollir un tamany adequat per a ell. El valor per defecte es de 50 elements per pàgina, però pot escollir entre 50, 100, 200, 500 o tots els elements per pàgina.

4.5 Instal·lació

El servidor és una màquina virtual que es troba dins d'un computador amb altres màquines virtuals. Per accedir a aquest servidor s'ha de fer mitjançant una connexió remota.

En aquesta màquina virtual (servidor) s'ha creat la base de dades tal com s'especifica al punt 4.2, s'ha creat l'usuari i s'han introduït les dades mitjançant els scripts mencionats a l'apartat 4.2.3.

Per últim s'ha copiat l'eina web i s'ha comprovat el correcte funcionament.

5 RESULTATS

A continuació es pot veure una imatge del resultat final.

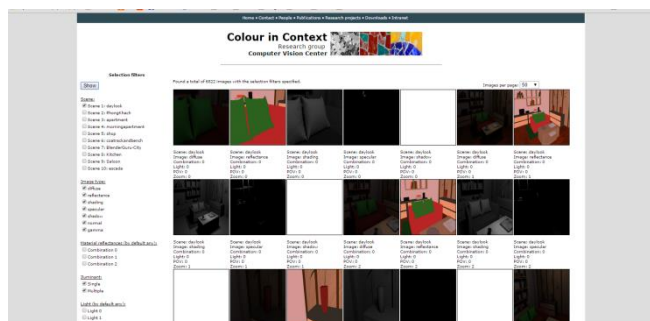


Figura 12. Aplicació web.

En les següents seccions es van explicant les funcionalitats de cada part.

5.1 Menús de selecció

El menú lateral de l'esquerra carrega el contingut de la base de dades per mostrar els filtres de selecció. D'aquesta manera, si en el futur s'expandís la base de dades, l'eina web automàticament mostraria el nou contingut.

Selection filters

[Show](#)

Scene:

☐ Scene 1: daylook

☐ Scene 2: PhongKhach

☐ Scene 3: apartment

☒ Scene 4: morningapartment

☐ Scene 5: shop

☐ Scene 6: coatrackandbench

☐ Scene 7: BlenderGuru-City

☐ Scene 8: Kitchen

☐ Scene 9: Saloon

☐ Scene 10: escada

Image type:

☒ diffuse

☒ reflectance

☒ shading

☒ specular

☒ shadow

☒ normal

☒ gamma

Material reflectances (by default any):

☐ Combination 0

☐ Combination 1

☐ Combination 2

Illuminant:

☒ Single

☒ Multiple

Figura 13. Menú lateral.

Donat la grandària d'aquest menú es disposa de dos botons per mostrar les imatges el principi i al final. Estan agrupats per context, és a dir, s'agrupen les opcions d'il·luminant, les opcions de les escenes, etc. de forma ordenada.

Es controla sempre que es seleccioni una escena, en cas de no seleccionar cap, dona un missatge d'error.

Selection filters

Please choose any selection filter in the left menu.

[Show](#)

Scene:

☐ Scene 1: daylook

☐ Scene 2: PhongKhach

☐ Scene 3: apartment

☐ Scene 4: morningapartment

☐ Scene 5: shop

☐ Scene 6: coatrackandbench

☐ Scene 7: BlenderGuru-City

☐ Scene 8: Kitchen

☐ Scene 9: Saloon

☐ Scene 10: escada

Figura 14. Missatge d'error que dona l'eina quan no es selecciona cap escena per mostrar.

5.2 Àrea de visualització

L'eina web s'adapta al contingut, expandint el seu tamany automàticament en base a la quantitat de contingut a mostrar (veure figures 9 i 10 de l'apartat 4.3.4)

Les imatges es visualitzen amb un tamany predefinit de 150 x 150 píxels agrupats en 7 columnes. D'aquesta manera es garanteix la integritat del disseny independentment del resultat a mostrar. Sota cada imatge es mostra un petit text amb informació dels filtres de selecció triats.



Figura 15. Agrupació per columnes dels resultats.

5.3 Paginació de resultats

L'eina fa ús d'un sistema de paginació que permet carregar de forma òptima les imatges sense crear una càrrega contundent en el servidor ni en la plataforma des d'on s'executa l'aplicació web. Es faciliten accessos directes a la següent pàgina o la anterior.

Depenent del nombre de resultats a mostrar es creen més o menys pàgines i es disposa d'enllaços per passar directament a alguna d'aquestes pàgines.

[\[Previous\]](#) [Page 72 of 137](#) [\[Next\]](#)

Showing 50 images per page.

[\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#) [\[6\]](#) [\[7\]](#) [\[8\]](#) [\[9\]](#) [\[10\]](#) [\[11\]](#) [\[12\]](#) [\[13\]](#) [\[14\]](#) [\[15\]](#) [\[16\]](#)

Figura 16. El botó Previous torna a la pàgina d'abans mentre que el botó Next ho fa per la següent. En casa de ser la primera pàgina el botó Previous no funciona i en cas de ser la última pàgina el botó Next és el que no funciona. A sota es poden veure les pàgines per poder accedir directament a alguna en concret.

Un altre element optimitzador és el ListBox de selecció per al tamany de pàgina el qual permet a l'usuari establir la quantitat d'elements a mostrar en la consulta. D'aquesta manera, per defecte, es mostren 50 elements ja que així el cost computacional per carregar és menor i la càrrega d'imatges és més ràpida. Si l'usuari vol visualitzar moltes imatges en poques pàgines, l'eina li permet fer-ho a expenses del rendiment. Aquest listbox també dóna l'opció de mostrar tots els elements en una única pàgina.

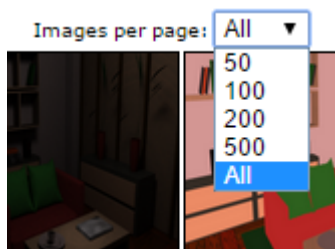


Figura 17. Listbox de selecció on l'usuari pot establir la quantitat d'imatges a mostrar en una única pàgina

5.4 Visualització vs temps

Es van fer tres comprovacions per comprovar que realment era necessari implementar un sistema de paginació i el Lazy Load. En tots el casos, l'ordinador era el client i el servidor alhora i la consulta es va fer des d'un ordinador amb processador Intel® Core(TM) 2 Duo a 2,1 GHz, 4 GB de ram i tarja de vídeo dedicada de 512 MB

La primera comprovació va ser sense utilitzar el plugins de Lazy Load, sense fer cap tipus de paginació. El resultat era que al carregar el que hi havia al viewport es veia bé, però l'ordinador trigava a respondre. Sobretot quan es desplaçava el viewport cap als resultats finals. A més, les últimes imatges no carregaven.

La segona comprovació va ser utilitzant la paginació, sense fer ús de Lazy Load i des del mateix ordinador. La millora va ser dràstica. Al limitar a 500 elements com a màxim els resultats per pàgina l'ordinador respon molt més ràpid que abans.

La tercera comprovació i actual estat de l'aplicació web és amb la paginació i el Lazy Load implementats. Com s'ha explicat abans, Lazy Load només carrega les imatges quan hi són al viewport. El mateix ordinador respon pràcticament al moment quan fa consultes i es desplaça el viewport cap al final dels resultats.

Per comprovar els resultats d'optimització es fan diferents proves de rendiment. A continuació es pot veure una taula amb els temps.

Taula 2. Taules de temps. Aquests càlculs es van fer carregant sempre la mateixa quantitat d'imatges (68220).

Paginació	Lazy Load	Elem./pàg.	Temps (segons)
No	No	68220	94,4 (*)
No	Sí	68220	47,4
Sí	No	50	5,2
Sí	No	100	5,4
Sí	No	200	5,5
Sí	No	500	6,5
Sí	Sí	50	4,8
Sí	Sí	100	5,3
Sí	Sí	200	5,5
Sí	Sí	500	6,2

(*) Les últimes imatges no arribaven a mostrar-se quan es baixava el viewport a baix del tot.

Cal tenir en compte que Lazy Load carrega la imatge quan aquesta és al viewport, per tant al fer scroll cap avall amb Lazy Load tot es més fluid i no dóna sensació de saturació. Amb grans quantitats d'imatges és on es veu millor l'efecte positiu que aporta aquest plugin.

5.5 Integració visual amb la resta de la web

El disseny de l'eina web utilitza el mateix estil que la resta de la web. Utilitza CSS per definir els estils.

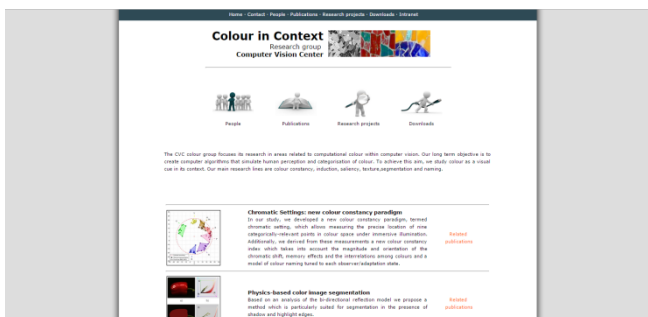


Figura 18. Web de l'equip de color.

L'eina web és una mica més ampla per donar espai a les imatges. Apart d'això, l'estètica és la mateixa. Es poden trobar els mateixos enllaços a la part de dalt, un peu de pàgina amb la mateixa informació i també els mateixos enllaços.

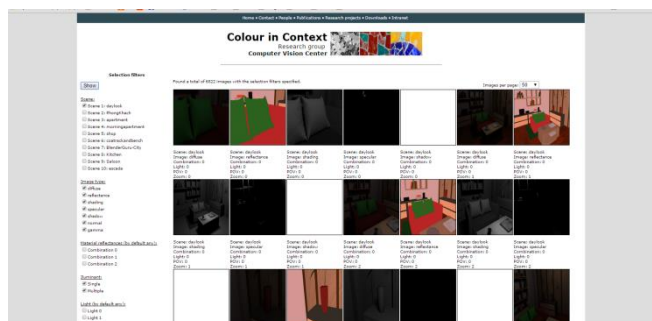


Figura 19. Aplicació web.

5.6 Seguretat

La principal característica de seguretat que té, com ja s'ha mencionat en l'apartat 4.2, és que l'usuari que accedeix a la base de dades només té permisos per consultar informació en la base de dades de l'eina. Si algú amb intencions dolentes volgués modificar alguna dada o fer algun mal a la base de dades a partir d'aquest usuari no podria fer-ho.

Per altre banda, la connexió a la base de dades, seguint unes "best practices", es fa en un fitxer diferent que es crida cada cop que es vol fer una consulta. Donat això, al ser codi PHP s'executa a la banda del servidor, no del client per la qual cosa no es pot visualitzar des del navegador. No hi ha una seguretat del 100%, però en tot cas, si volguessin fer qualsevol tipus de mal, necessitarien la contrasenya de root del servidor, cosa que des de l'eina no es pot obtenir de cap de les maneres.

6 CONCLUSIÓ

El projecte està finalitzat complint amb els requisits del client. S'ha dissenyat la base de dades, la pàgina web i s'ha instal·lat al servidor del client.

Un cop finalitzat el projecte, s'ha obtingut una aplicació web totalment funcional que permet escollir els filtres de selecció desitjats i obtenir resultats de forma eficient.

Sobretot, el més important que el que s'ha fet, és com s'ha fet: la paginació i la implementació d'AJAX i el plugin Lazy Load. Aquest dos són punts clau en el desenvolupament que han determinat la bona optimització de l'aplicació.

Un pla de futur inclou millorar el disseny del menú lateral aplicant estàndards d'usabilitat al filtre de "Light". En comptes de tenir una llista amb tots els elements d'aquests atribut, fer dues llistes del 0 al 5 i que l'usuari faci combinacions d'aquestes dues llistes per obtenir els resultats.

També seria interessant agrupar els resultats de les consultes mostrant com a molt sis columnes i fer ordenacions dels resultats per tenir agrupats els continguts de les escenes per atributs. D'aquesta manera es té una ordenació dels resultats més polida.

Per últim, caldria controlar els checkboxes seleccionats i fer que segons la selecció que s'hagi fet alguns checkboxes no siguin seleccionables.

La meua aportació ha sigut la de crear l'eina i la base de dades a la qual està lligada. A més aquesta nova base de dades pot ser utilitzada per qualsevol altre fi i l'estructura permet fer tasques de les que es pot esperar un bon resultat.

Aquest projecte m'ha servit per aprendre com implementar AJAX i jQuery a una pàgina PHP i fer ús de la

paginació dels resultats. Aquests coneixements són molt útils ja que avui en dia s'utilitzen en molts àmbits com son xarxes socials, pàgines de e-commerce, etc.

AGRAÏMENTS

Estic molt agraït de Robert Benavente per tota l'ajuda i orientació facilitada en la realització d'aquest projecte, per els consells en el disseny de l'eina i els consells sobre els documents i informes presentats.

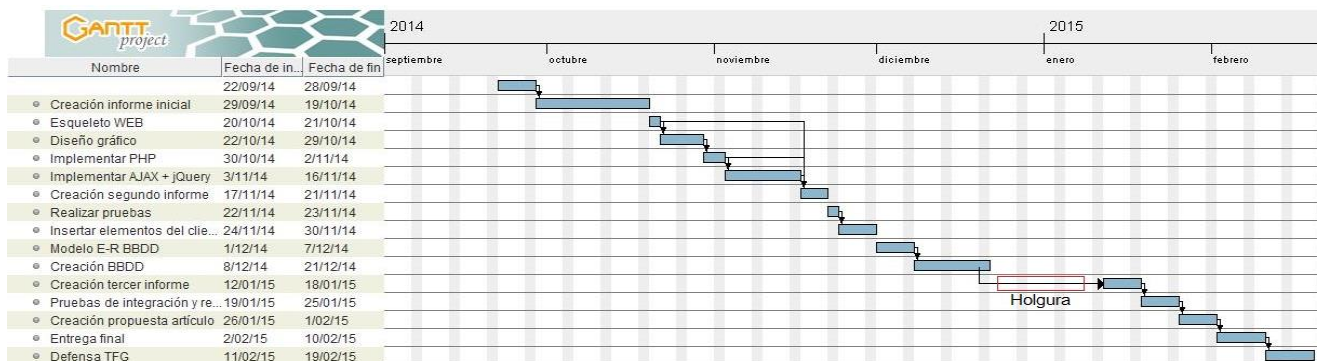
BIBLIOGRAFIA

- [1] Describable Textures Dataset (DTD), Internet: <http://www.robots.ox.ac.uk/~vgg/data/dtd/>, [Última data d'accés: Setembre, 22, 2014]
- [2] A. Jiménez, «CREACIÓN DE UNA BASE DE DATOS DE IMÁGENES SINTÉTICAS PARA LA EVALUACIÓN DE ALGORITMOS DE ESTIMACIÓN DE IMÁGENES INTRÍNSECAS», Proyecto final de carrera, UAB, Juliol, 2014.
- [3] When and why are database joins expensive?, Internet: <http://stackoverflow.com/a/174047>, [Última data d'accés: Desembre, 8, 2014]
- [4] Matlab, El lenguaje de cálculo técnico, Internet: <http://es.mathworks.com/products/matlab/>, [Última data d'accés: Desembre, 13, 2014]
- [5] MySQL Documentation: MySQL Reference Manuals, Internet: <http://dev.mysql.com/doc/>, [Desembre, 15, 2014]
- [6] PHP Documentation, Internet: <http://php.net/docs.php>, [Última data d'accés: Octubre, 30, 2014]
- [7] Asynchronous JavaScript and XML (AJAX), Internet: http://www.w3schools.com/ajax/ajax_intro.asp, [Última data d'accés: Novembre, 7, 2014]
- [8] jQuery API Documentation, Internet: <http://api.jquery.com/category/ajax/>, [Última data d'accés: Novembre, 7, 2014]
- [9] Lazy Load Plugin for jQuery, Internet: <http://www.appelsiini.net/projects/lazyload>, [Última data d'accés: Novembre, 11, 2014]
- [10] What is the Document Object Model?, Internet: <http://www.w3.org/DOM/#what>, [Última data d'accés: Novembre, 7, 2014]

APÈNDIX

A1. PLANIFICACIÓ

En aquest apèndix s'inclou la planificació del projecte i es detallen els canvis introduïts en aquesta planificació al llarg del desenvolupament.



FA1. Diagrama de Gantt de la planificació. Es pot veure els marges de temps planificats per solventar punts crítics.

A1.1 Tasques

- Anàlisi de requisits i creació d'informe
 - Entrevista amb el tutor (client) per captar els requisits.
 - Documentar els requisits.
 - Estudiar tecnologia a utilitzar.
 - Documentar l'informe.
- Creació de l'eina web.
 - Crear l'esquelet web amb HTML.
 - Realitzar el disseny gràfic de la interfície web.
 - Implementar PHP.
 - Implementar AJAX i jQuery.
 - Implementar el plugin Lazy Load de jQuery.
- Creació de la base de dades.
 - Crear model Entitat-Relació.
 - Implementar el model en MySQL.
 - Preparar les dades estructurades en directoris.
 - Generar scripts d'introducció de les dades a MySQL.
- Proves d'integració.
 - Provar la integritat de la base de dades amb l'eina web.
 - Corregir errors.
 - Detectar punts crítics.
- Documentació del projecte i preparació de la defensa.
 - Creació dels informes de progrés.
 - Creació de l'article.
 - Preparar dossier amb tot el material.
 - Preparar la defensa del TFG.

A1.2 Canvis en la planificació

El desenvolupament del projecte ha patit un petit retràs en les dates donat a problemes que varen sorgir durant la realització de la creació de la base de dades.

No obstant, a la planificació es va considerar la possibilitat de trobar dificultats i es van establir marges de temps (veure apèndix FA1. Diagrama de Gantt) per que no afectessin a la resta de tasques i així poder seguir amb el calendari previst.

Per una altra banda, el desenvolupament es va avançar una mica respecte la planificació inicial al desenvolupar la part d'implementar AJAX. Aquest avanç es deu a que AJAX va molt lligat amb la llibreria jQuery ja que tots dos formen part de Javascript.

Aquest avanç ha proporcionat més marge per familiaritzar-se amb les tecnologies AJAX i jQuery sobre les quals no es tenia coneixements previs.

A més, les tasques d'integració ha sigut més natural de lo que s'havia previst, ja que en comptes de desenvolupar cada bloc per separat e integrar-los tots al final, s'han anat implementant de forma seqüencial.